Overture

Object-Oriented Tools for Solving CFD and Combustion Problems in Complex Moving Geometry

University of California
Lawrence Livermore National Laboratory

CASC Center for Applied Scientific Computing

Technology

Overture is an object-oriented code framework for solving partial differential equations in serial and parallel computing environments. It provides a portable, flexible software development environment for applications that involve the simulation of physical processes in complex moving geometry. It is implemented as a collection of C++ libraries that enable the use of finite difference and finite volume methods at a level that hides the details of the associated data structures, as well as the details of the parallel implementation. Overture is designed for solving problems on a structured grid or a collection of structured grids. In particular, it can use curvilinear grids, adaptive mesh refinement, and the composite overlapping grid method to represent problems involving complex domains with moving components.

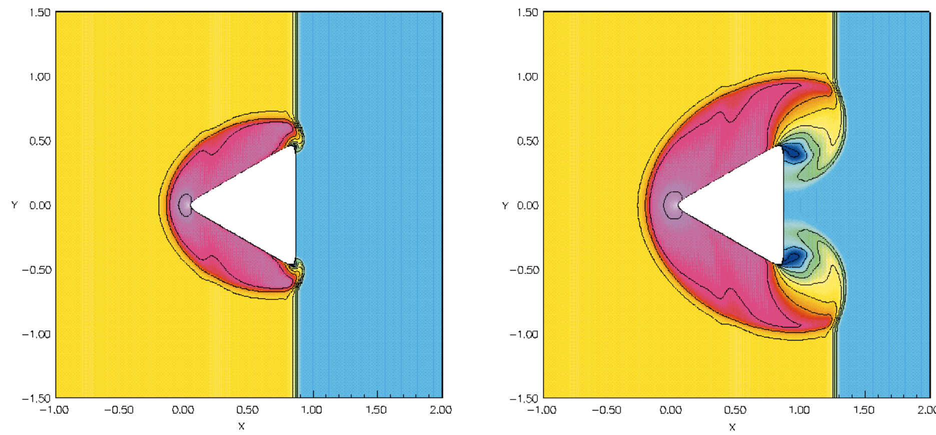

Figure 1. Simulating compressible flow past a rounded triangle is readily accomplished using Overture.

The method of adaptive composite overlapping grids provides a computational mechanism to accurately simulate physical processes that are described by systems of partial differential equations (PDEs) in complex moving domains. High-resolution finite difference or finite volume methods are used on a collection of structured curvilinear grids. This basic differencing technology is combined with block-structured adaptive mesh refinement (AMR) to provide local resolution in the computation with correspondingly greater computational efficiency. At Lawrence Livermore National Laboratory (LLNL), this approach is used for high-resolution simulations of incompressible and low Mach number hydrodynamics flows in complex moving geometries. The payoffs for using this method are significant, but the implementation details can be complex due to the complexity of both the overall algorithm and the physics behind the simulation.

The Overture Framework

While it is possible to use traditional structured programming approaches to implement, debug, modify and maintain applications codes based on the adaptive composite overlapping grid method, this would be a daunting task, particularly on parallel computers. The Overture framework was developed using object-oriented design techniques and with the C++ programming language. With object-oriented design, the task is to develop computational "objects" that represent fundamental abstractions of elements in a computational model. Where in the structured approach, the fundamental unit of code is a subroutine or function that modifies the data in some way, in the object-oriented approach, the fundamental unit is an object described by a class in C++. A class contains both data that describe the object, and functions that operate on those data.

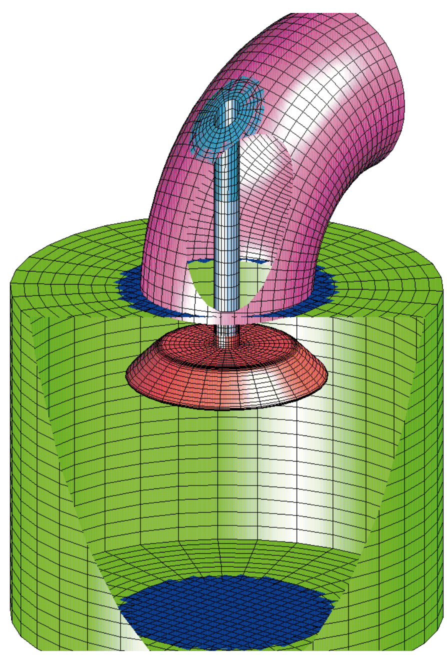

Figure 2. Overlapping grids can be used to represent complex geometries by combining component grids. This overlapping grid represents an engine valve, port and cylinder.

In Overture, object-oriented design principles have been used throughout to hide the details of complex data structures and algorithms and their parallel implementation. Data structures and algorithms can be specialized and extended through derivation. The Overture classes provide tools for the rapid development of application codes. The main class categories are listed below.

1. *A++/P++ arrays* describe multidimensional arrays and provide for serial and parallel operations on those arrays. In the parallel envi-

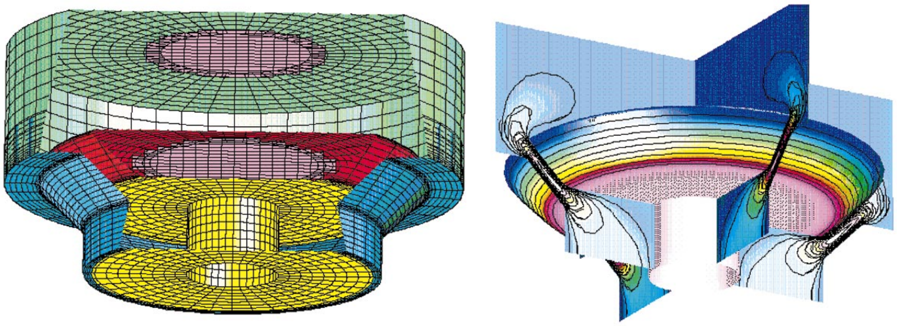

Figure 3. Overlapping grid and computational fuid dynamics solution for a moving 3-D valve. The image on the left shows a cut-away view of the component grid surfaces for the valve, valve seat and the flow chamber. The figure on the right shows an incompressible flow solution in this geometry. Contours of the pressure are plotted on the surface of the valve; contours of velocity magnitude are plotted on cutting planes through the flow region.

ronment, these provide for the distribution and interpretation of communication required for the data parallel execution of operations on the arrays.

2. *Mappings* define transformations such as curves, surfaces, areas and volumes. These are used to represent the geometry of the computational domain.
3. *Grids* define a discrete representation of a mapping or mappings. These include single grids and collections of grids, in particular composite overlapping grids. The *Ogen* overlapping grid generator provides tools for the construction of curvilinear grids, and for overlapping those grids to represent complex moving geometries.
4. *Grid functions* provide for the representation and centering of solution values such as density, velocity and pressure, defined at each point on the grid(s).
5. *Operators* provide discrete representations of differential operators and boundary conditions through finite difference or finite volume approximations.
6. *Visualization tools* based on OpenGL are provided to furnish a high-level graphics interface for visualizing geometry and simulation results.
7. *Adaptive mesh refinement* provides automatic refinement of the overlapping grid structure for increased local resolution and efficiency of computational simulations.
8. *Load-balancing tools* are provided for automatic load-balancing of computations on the adaptive overlapping grid structure on parallel computers
9. *Parallel distribution mechanisms* are provided through the PADRE library, part of the DOE 2000 ACTS toolkit.

Overture Application Code Development

In collaboration with scientists at LLNL, Los Alamos National Laboratory, and in academia, Overture is being used to develop flow solvers for high-speed compressible flow problems, incompressible flow problems, low Mach number and non-Newtonian fluid flow problems. In particular, the *OverBlown* flow solver for reacting flow applications supports the DOE Energy Research project on High-Resolution Computational Fluid Dynamics and Combustion Modeling for Diesel Engine Simulation in collaboration with scientists at Lawrence Berkeley Laboratory and New York University.

Through its object-oriented design, Overture reduces code duplication, encourages interoperability of application software, and simplifies the learning curve for new computational methods. Overture's object-oriented architecture provides flexibility to address a wide range of applications that involve simulations in complex moving geometry on serial and parallel computers. The advantages of this approach include reduced code development time and broader, more in-depth research into numerical methods for scientific and industrial applications.

For additional information about Overture, contact David L. Brown, (925) 424-3557, dlb@llnl.gov; or Daniel J. Quinlan, (925) 423-2668, dquinlan@llnl.gov; or William Henshaw, (925) 423-2697, henshaw@llnl.gov.

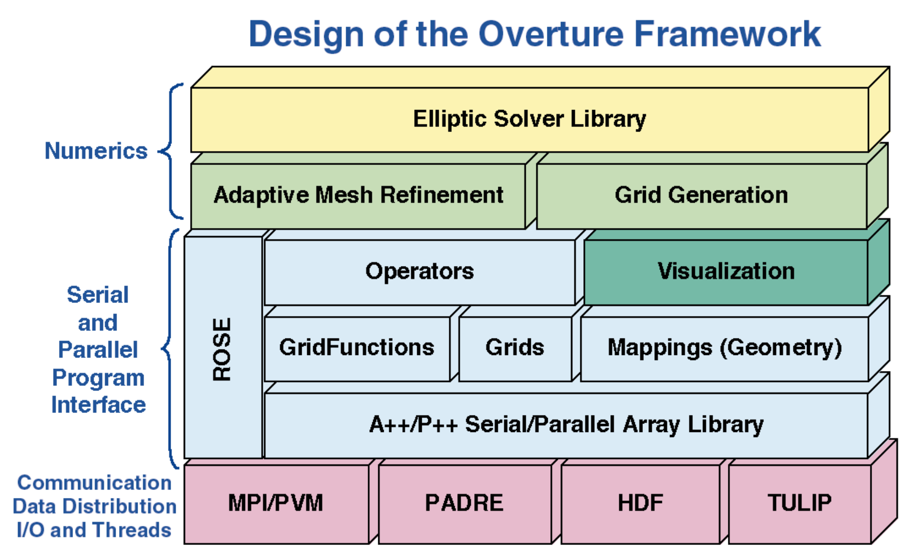

Figure 4. Hierarchichal design of the Overture framework. Tools at all levels in the Overture framework are available to application developers. Most commonly, the components of the Numerics and Serial and Parallel Program Interface levels would be used to develop partial differential equation solvers.

Work performed under the auspices of the U.S. DOE by LLNL under contract No. W-7405-ENG-48